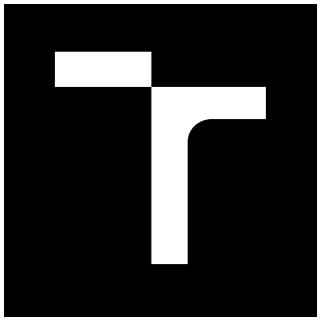
VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2018

Nataliya Golovkova



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

KOMUNIKACE MIKROTIK A IPS
COMMUNICATIONS MIKROTIK AND IPS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Nataliya Golovkova**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. Tomáš Gerlich**
SUPERVISOR

BRNO 2018

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Studentka: Nataliya Golovkova ***ID:*** 189046
Ročník: 3 ***Akademický rok:*** 2017/18

NÁZEV TÉMATU:

Komunikace MikroTik a IPS

POKYNY PRO VYPRACOVÁNÍ:

Cílem bakalářské práce je implementace komunikačního protokolu mezi směrovači Mikrotik a IPS (Intrusion Prevention System) systémem Suricata. Cílem této komunikace je zmírnění dopadu útoků, které jsou cíleny na odepření služby webového serveru (DoS - Denial of Service). V rámci bakalářské práce realizujte návrh a implementaci komunikačního protokolu na experimentálním pracovišti VUT v Brně. Implementovaný komunikační protokol otestujte a dosažené výsledky přehledně zpracujte. Hlavním výsledkem bakalářské práce bude funkční řešení komunikačního protokolu.

DOPORUČENÁ LITERATURA:

[1] AKBAR, Salman; ENDROYONO, Adhi Dharma Wibawa. The Impact Analysis of DDoS Attacks on Government Electronic Procurement Service (LPSE) and Mitigating DDoS Attacks Using Intrusion Detection System and Honeypot.

[2] BHOSALE, Dhanashri Ashok; MANE, Vanita Manikrao. Comparative study and analysis of network intrusion detection tools. In: Applied and Theoretical Computing and Communication Technology (iCATccT), 2015 International Conference on. IEEE, 2015. p. 312-315.

Termín zadání: 5.2.2018 ***Termín odevzdání:*** 29.5.2018

Vedoucí práce: Ing. Tomáš Gerlich
Konzultant:

prof. Ing. Jiří Mišurec, CSc.
předseda oborové rady

ABSTRAKT

Bakalářská práce je zaměřena na problematiku síťových útoků a případnou ochranu před jejich následky. V teoretické části byly popsány útoky, které patří v současné době mezi nejvíce rozšířené. Především se zabývá analýzou útoků pro odepření služeb. V další části práce jsou rozebrány detekční a preventivný systémy pro monitorování síťového provozu, s důrazem na systém Suricata. V následující část slouží k seznámením se zařízeními firmy Mikrotik, která jsou využívána v praktické části práce. V praktické části je řešena komunikaci mezi detekčním systémem Suricata a routerem Mikrotik, pro zajištění mitigace DoS útoků. Komunikace je řešena formou skriptu pomocí programovacího jazyku php.

KLÍČOVÁ SLOVA

DoS, DDoS, IDS, IPS, Mikrotik, Suricata, RouterOS

ABSTRACT

The bachelor thesis is is focused on network attack problems and possible protection against their consequencies. The theortecical part describes the attacks that are currently among the most widespread with focus on the attack of Denial of Services (DoS). The next part of the thesis deals with detection and prevention systems fornetwork traffic monitoring with emphasis on the Suricata system. The following part is about getting familiar with the Mikrotik devices that are used in the practical part of the thesis. The practical part aims to provide a solution to mitigate the DoS attack in the communication between Mikrotik router and Suricata system. The communication is solved in a script using the php programming language.

KEYWORDS

DoS, DDoS, IDS, IPS, Mikrotik, Suricata, RouterOS

GOLOVKOVA, Nataliya. *Komunikace MikroTik a IPS*. Brno, 2018, 49 s. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce: Ing. Tomáš Gerlich,

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Komunikace MikroTik a IPS“ jsem vypracovala samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autorky

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu diplomové práce panu Ing.Tomáš Gerlich za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno ....

podpis autorky

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
http://www.six.feec.vutbr.cz

PODĚKOVÁNÍ

Výzkum popsaný v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno ....

podpis autorky

Obsah

Seznam obrázků

Úvod

Tato bakalářská práce se věnuje k zajištění komunikace IDS suricaty a zařízení Mikrotik. Popisuje, jak lze chránit počítačovou síť a jak fungují tyto bezpečnostní systémy. V práci jsou podrobněji zváženy systémy detekce, detekce narušení (IDS), jejich typy a princip ochrany proti útokům z internetu.

Práce popisuje využití jednotlivých typů IDS. To vysvětluje, proč některé typy systémů detekce narušení jsou preferovány ostatními. Dále jsou zde uvedeny vady a nedostatky různých typů IDS.

Kvůli nárůstu počtu útoků v jakékoli síti je potřeba provést bezpečnost v systému. Data musí být chráněny, čímž lze zabránit jejich ztrátě či zneužití. Tato bakalářská práce také popisuje nejčastější útoky, které se mohou vyskytnout i jejich dopad na počítačovou síť.

Dále je v práci popsáno, jak může správce sítě rozpoznat určité typy internetových útoků. Lze také zkoumat možnosti spolupráce mezi operačním systémem RouterOS a preventivním programem Suricata, který je široce používán jako IDS.

Praktická část práce se zabývá implementací skripty která posílá do Mikrotiku balíček s IP adresou. Tento systém je testován SYN a UDP útoky.

1 Klasifikace síťových útoků

Hlavním cílem je většinou při každém útoku získání neautorizovaný přístupu k informacím. Útoky na koncová zařízení nebo sítě mohou být klasifikovány podle následujících charakteristik.

1.1 Pasivní

Pasivním je nazýván útok, v němž nepřítel neschopen upravit přenášené zprávy a vložit své zprávy do informačního kanálu mezi odesílatele a příjemce. Účelem pasivního útoku může být pouze odposlouchávání vyslaných zpráv a analýza za provozu. V tomto případě je obecně dáno, že je porušena důvěrnost.
Jedním z typů pasivních útoků je skenování útoků. Vznikají odesláním různých typů paketů v době, kdy útočník sleduje cílovou síť nebo systém.
Při analýze odpovědí se útočník může hodně naučit o charakteristice a zranitelnosti systému. Útok skenováním je pro útočníka prostředkem identifikace cíle.
Použité nástroje mají různá jména, např. síťové analyzátory, analyzátory portů, síťové skenery, skenery portů nebo snímače zranitelnosti. Skenování útoků může určit:

- typologie cílové sítě;
- typy síťové komunikace, které jsou předávány prostřednictvím brány firewall;
- aktivní hostitelé v síti;
- operační systémy běžící na počítačích;
- serverový software, který běží na hostitelích;
- čísla verzí pro veškerý detekovaný software.

Skenery pro zranitelnost jsou speciální typ skeneru, který kontroluje konkrétní chyby na hostitelích. Útočník může spustit skener pro zjištění zranitelnosti, jenž detekuje seznam hostitelů (adresy IP), které jsou zranitelné vůči určitému útoku.
S těmito informacemi může útočník přesně určit různé parametry softwaru oběti tak, aby je použil k provádění konkrétních útoků, pak také aby mohl proniknout do těchto systémů. Jinými slovy lze říci, že útočníci používají skenování k výběru cíle před spuštěním skutečného útoku. Analogicky s tím, že kdokoli může vstoupit do banky či prověřit viditelný bezpečnostní systém.
Hlavní věc je, že zpravidla technologie, která umožňuje detekovat veřejně dostupné zdroje, také umožňuje analyzovat systém, aby našel slabé stránky v zabezpečení. Je-li síť připojena k internetu, téměř bude skenována.[15]

1.2 Aktivní

Aktivní útok má přímý dopad na fungování samotného systému, který může porušit bezpečnostní politiku. Ve srovnání s pasivní funkcí je aktivní funkce akcí, jejíž hlavní možností je detekce a v důsledku její implementace systém prochází určitými změnami. Zde je uveden seznam některých aktivních útoků, které jsou nejčastěji používané na síti:

Packet sniffers

Packet sniffer je aplikační program, který používá síťovou kartu, jež zachycuje všechny síťové pakety i přenos přes konkrétní doménu.

IP spoofing

IP spoofing se týká falšování zdrojové IP adresy a její následné odeslání k cílovému počítači, před kterým chceme svou skutečnou IP adresu udržet v tajnosti. To lze provést dvěma způsoby. Za prvé hacker může použít adresu IP, v rozsahu autorizovaných adres IP nebo autorizovanou externí adresu, do které je povolen přístup určitého síťového zdroje. IP spoofing útoky jsou často výchozím bodem pro další útoky. Klasickým příkladem je útok DoS, který začíná adresou někoho jiného. Ta ovšem skrývá pravou identitu hackera.

Útoky typu Man-in-the-Middle

Při útoku typu Man-in-the-Middle potřebuje hacker přístup k paketům přenášených přes síť. Takový přístup ke všem paketům přenášeným z poskytovatele do jiné sítě může být například získán zaměstnancem tohoto poskytovatele. Pro tento typ útoku se často používají paketové snifery, transportní i směrovací protokoly.

Útoky na úrovni aplikace

Útoky na úrovni aplikací mohou probíhat několika způsoby. Nejběžnější je použití známých nedostatků serverového softwaru (sendmail, HTTP, FTP).

1.3 Velké množství pingů

Snadný způsob jak zjistit které počítače jsou v síti , je masové spouštit příkaz ping. Zařízení která odpoví na tento dotaz, útočník tak schopen prozkoumat potenciální cíle útoku. Velice efektivní chránění je zakázání odpovědí na tento požadavek na firewallu, poté náš systém bude vykazovat útočníkovi jako vypnutý. Jestliže si můžeme na firewallu zvolit si akcí použitých na sledovaný provoz, musíme rozpoznat akce REJECT a DROP.[6] Akce REJECT sice způsobí zahození paketu, ale pošle zpět ICMP zprávu o nedostupnosti.[6] Pro obranu proti skenování z toho důvodu aplikujeme akce DROP, ta zahodí paket bez oznámení útočníka ICMP zprávou.

2 Zabezpečení síťových zařízení

Cíl síťových útoků je ve většině případů koncové stanice. Nemůžeme taky zapomenout na to, že zranitelná jsou i samotná síťová zařízení, u kterýkoli může útočník nadělat zničující poruchy. Hlavním cílem je, aby se zabránilo potenciální útočníky od vlivu na funkčnost sytových prvků. Útoky můžou mít hodně form, od velkého vytížení zařízení, který nedovolí zařízení správně provádět provoz, až po odstranění zařízení.

2.1 Chránění zařízení

Představa o ochraně zařízení, je zajistit, že funkčnost zařízení není přerušena působením okolí. Z toho důvodu musíme přijmout takové opatření a pravidla k zabezpečení provozu zařízení, aby nedocházelo k porušení. Jedním z předpokladů je umístění ve specializovaném místě. Kde bych především byla klimatizace, kvalitní a zálohovaná elektrická síť.

2.2 Obnovení operačních systémů

Jako téměř u každého operačního systémů tak i u softwaru síťových zařízení můžeme zjistit chyby. Tyto chyby mohou být jiné povahy. Jestliže je chyba nalezena a oznámená výrobci software je ve většině případů v další verzi operačního systému bude opravena. Z tohoto důvodu vyplývá, že je vhodné udržovat OS na zařízení aktuální. Mezi základní pravidla patří zálohování operačního systému a to je možno s výhodou použít při hledání nevhodné konfigurační úpravy. V případě selhání zařízení potřebujeme také zálohování. OS může být bezchybné, ale stále mohou podlehnout DoS úroků, kdy útočník vyčerpá všechny zdroje zařízení (procesor, paměť, disková kapacita).[6]

Aby nedošlo k beznadějné situaci během závalu DDoS v systémech, je třeba je na tuto situaci pečlivě připravit:
Všechny servery, které mají přímý přístup k externí síti, musí být připraveny pro jednoduché a rychlé vzdálené restartování počítače. Velkým přínosem je přítomnost druhého administrativního síťového rozhraní, díky němuž je přístup k serveru v případě přerušení provozu hlavního kanálu.
Software používaný na serveru musí být vždy aktuální. Tím jsou všechny zranitelnosti opraveny, aktualizace nainstalovány (jednoduché jako boot, ovšem rada, kterou mnozí přehlížejí). To ochrání před útoky DoS a využíváním chyb ve službách.
Všechny služby poslechových sítí, které jsou určené pro administrativní použití, musí

mít brány firewall skryty od každého, kdo k nim nemá přístup. Útočník pak není schopen je použít k útoku DoS.
Na přístupech k serveru (blízko routeru) musí být nainstalován systém analýzy provozu, který umožní včas upozornit na začínající útok, a tedy včas přijmout opatření k jeho předcházení.
Je třeba poznamenat, že všechny triky jsou zaměřeny na snížení účinnosti těch DDoS útoků, jejichž cílem je vyčerpat prostředky počítače. Od záplavy, která zablokuje kanál s odpadem, je téměř nemožné bránit se a jediný správný, ale ne vždy praktický způsob boje je „zbavit útok významu“. Pokud je kanál opravdu široký, pak je možné, že od 90 % útoků je server chráněn.

3 Systém detekce a prevence průniku

IDS - je zařízení nebo software, který monitoruje činnost sítě a systému a zjišťuje škodlivé akce a porušování zásad zabezpečení. Existují dva hlavní typy IDS – síťová a uzlová (další jsou popsány níže). Hlavní úkoly systémů detekce a prevence narušení jsou detekce možných incidentů, ukládání informací o nich a hlášení pokusů o narušení. Služba IDS/IPS zpravidla zaznamenává informace přímo související s pozorovanými událostmi, upozorní správce sítě v případě potenciálního narušení a sestavuje zprávu o událostech. Mnoho IDS/IPS také může odolat zjištěným hrozbám a snaží se zabránit tomu, aby byl útok úspěšný. Samotný IDS/IPS kupříkladu může zastavit útok, změnit nastavení brány firewall, nebo změnit obsah útoku. [1] [10]

3.1 Struktura detekce průniku

Subsystém senzorů, který shromažďuje informace týkající se zabezpečení sítě. Koncentrátor, který uchovává informace získané ze snímačů a informace zpracované analyzátorem. Analyzátor, který detekuje podezřelou komunikaci a útoky na základě dat ze senzorů. Konzola pro správu, která umožňuje konfiguraci IDS.

3.2 Klasifikace detekce průniku

Síť IDS je obvykle zařízení nainstalované v síti. Uzlové IDS obvykle představují program – klient a jsou instalované na zařízení v síti. IDS orientované na protokoly – systémy, které analyzují data přenášená určitými protokoly. Hybridní IDS kombinují dva nebo více přístupů.

HIDS (host-based intrusion detection system) Je umístěn na samostatném uzlu a sleduje příznaky útoků na tomto uzlu.

Uzel IDS (HIDS) je systém senzorů, které jsou načteny na různých serverech organizace a řízeny centrálním dispečerem. Senzory sledují různé typy událostí a provádějí určité akce na serveru nebo přenášejí oznámení. HIDS senzory sledují události spojené se serverem, na kterém jsou načteny. HIDS senzor umožňuje určit, zda byl útok úspěšný, pokud k útoku došlo na stejné platformě, na které byl snímač nainstalován.

Základní typy senzorů HIDS.

Analyzátor záznamu - proces je spuštěn na serveru a sleduje příslušné soubory záznam v systému. Pokud záznam odpovídá kritériu v procesu snímače HIDS, provede se nastavená akce.

Analyzátory záznamu logu nebrání útoku v systému, ale reagují na událost poté, co

k ní došlo.

Senzory označení(indikace)

Jedná se o sady určitých atributů událostí zabezpečení, které jsou mapovány na příchozí přenosy nebo záznamů v logu. Možnost analýzy příchozího provozu, je rozdílem snímače dat z analyzátorů záznamu logu.

Příznaky senzoru HIDS je užitečná při sledování přihlášené uživatele v rámci informačních systémů.[16]

Analyzátory systémových volání

Tyto analyzátory analyzují volání mezi aplikacemi a operačním systémem k identifikaci událostí souvisejících se zabezpečením. HIDS snímače tohoto typu umístí softwarový hrot mezi operačním systémem a aplikacemi. Když aplikace provede akci, jeho volání operačního systému je analyzován a mapována na databázi, které jsou příklady různých typů chování, které představují, útočné akce nebo objekt, který je předmětem zájmu pro správce.

Analyzátory systémových volání se liší od výše uvedených senzorů, takže mohou zabránit akcím.

Kontrola celistvosti souborů

Sleduje změny v souborech pomocí kryptografického kontrolního součtu nebo digitální podpis souboru (Šifrování). Když změníte i jen malé části zdrojového souboru (to může být jak atributy souboru, jako je čas a datum vytvoření), konečný digitální podpis souboru bude změněn. Cílem tohoto algoritmu je minimalizovat možnost změny souboru při zachování předchozího podpisu.[16]

Rozdíl ukazuje, že soubor byl změněn.Kontrola integrity neidentifikuje útok, ale podrobně popisuje výsledky útoku.

(network intrusion detection system) Nachází se na samostatném systému, který monitoruje síťový provoz pro příznaky útoků prováděných v řízeném segmentu sítě.

Systém detekce narušení sítě (NIDS) je softwarový proces spuštěný ve vyhrazeném systému a je zodpovědný za přepnutí síťové karty do systému do nečitelného režimu provozu, při kterém síťový adaptér přenáší veškerý síťový provoz do softwaru NIDS. Analyzuje provoz pomocí souboru pravidel a atributů útoku, aby zjistil, zda tato návštěvnost představuje nějaký zájem. Poté je generována odpovídající událost.

V současné době má většina NIDS systémů sadu atributů útoku, s nimiž je mapována komunikace v komunikačním kanálu. Při neexistenci jakýchkoli příznaků útoku v systému detekce narušení, systém NIDS tento útok nezaznamená. Tyto systémy umožňují zadat sledovanou návštěvnost na zdrojové adrese, cílové adrese, zdrojovém portu nebo cílovém portu. Umožňuje sledovat provoz, který neodpovídá charakteristice útoků.

Výhody NIDS:

- NIDS může být v síti zcela skryto takovým způsobem, že útočník nebude vědět, že je sledován;
- Jeden systém NIDS může být použit pro sledování provozu s velkým počtem potenciálních cílových systémů;
- NIDS může zachytit obsah všech paketů určených pro cílový systém;

Nevýhody NIDS:

- Systém NIDS může vydávat poplach pouze tehdy, pokud se provoz shoduje s předdefinovanými pravidly nebo charakteristikami;
- NIDS může kvůli využití široké šířky pásma nebo alternativních cest vynechat požadovaný provoz;
- Systém NIDS nemůže určit, zda byl útok úspěšný;
- Systém NIDS nemůže vidět šifrovanou komunikaci;
- V komutovaných sítích (na rozdíl od sítí se sdílenými médii) jsou vyžadovány speciální konfigurace, bez kterých NIDS nebude kontrolovat veškerou komunikaci.

3.3 Sítove a uzlove systém detekce/prevence průniku

3.3.1 Systémy detekce vniknutí do sítě

Síťové IDS jsou obvykle instalovány na strategických místech uvnitř sítě a zpravidla jsou to zařízení připojená k síti organizace. Analyzují veškerou komunikaci, která prochází celou podsítí, pracují v nečitelném režimu, a poté porovnávají průchodnou návštěvnost do databáze již známých útoků. Jakmile je detekován a identifikován útok, administrátorovi je zasláno oznámení. Příklad konfigurace síťové IDS je jeho instalace do jediné podsítě s firewally, aby byly zjištěny pokusy obejít je. V ideálním případě síťové IDS kontroluje veškerý příchozí a odchozí provoz, ale v praxi to může vést k „překážkám“, čímž se sníží výkonnost sítě jako celku.

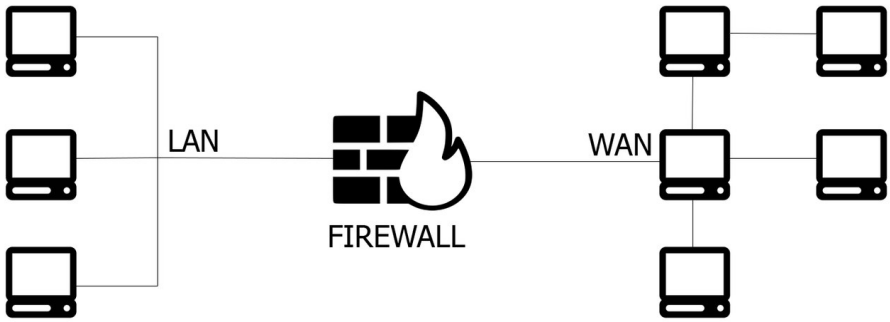
3.3.2 Uzlový systém detekce narušení (IPS) bezpečnosti

Systémy IPS lze považovat za rozšíření systémů detekce (IDS), jelikož úloha sledování útoků zůstává stejná. Nicméně se liší v tom, že IPS by měl sledovat činnost v reálném čase a rychle provádět opatření k předcházení útokům. Systémy detekce vniknutí uzlu fungují na samostatných zařízeních a pracovních stanicích sítě. Uzlová IPS analyzuje provoz pouze jednoho zařízení a varuje administrátora či uživatele

v případě poplachu. Uzlová IPS navíc při instalaci obvykle vytváří zálohu systémových souborů a pravidelně je porovnává se současným stavem těchto souborů. V případě změny nebo nepřítomnosti okamžitě oznámí správci následné vyšetření situace. Tento systém je často instalován na kritických místech, které nezabezpečují změnu nastavení systému.

3.3.3 Srovnání s firewallem(Firewall)

Přestože jsou firewally i IDS systémy určené k zabezpečení sítě, existují mezi nimi značné rozdíly. Hlavním úkolem firewallů je filtrování provozu mezi sítěmi jak na Obr. 3.1, aby se zabránilo útokům. Typické použití firewallu je na kraji sítě kde tvoří rozhraní mezi LAN sítí a WAN sítí. Následující funkce je záznam probíhající komunikace a možná modifikace procházejících paketů. Firewall můžeme spravovat na routeru nebo na hardwaru. Ovšem brány firewallů nemohou správce upozornit na útok nebo podezřelou aktivitu v síti. Systémy detekce narušení jsou neustále sledovány, obcházejí veškerý provoz jen přes firewall, potenciální porušení bezpečnostní politiky zevnitř i uvnitř sítě a o všem informují operátory. Toho se obvykle dosahuje kombinací technik, např. Analýzou síťové komunikace, heuristickou analýzou a identifikací signatur známých útoků a jejich okamžité oznámení.



Obr. 3.1: Zapojení firewallu

3.4 Hlavní metody analýzy používané IDS

Dva hlavní přístupy k analýze síťové aktivity jsou statistické a podpisové. Moderní systémy pro odhalení průniku obvykle používají kombinaci obou těchto metod.

3.4.1 Statistické systém detekce průniku

Systémy pro odhalení průniku, které používají statistický přístup po instalaci, jsou „školeny“ administrátorem, který nastavuje zásady IDS odpovídající normální aktivitě v síti – typy provozu, připojení mezi uzly, protokoly a porty. Pokud jsou v síti zjištěny anomálie nebo statisticky významné rozdíly v provozu z typické sítě, IDS o tom informuje správce. Hlavním problémem tohoto přístupu je obtížnost nastavení a velký počet falešně pozitivních poplachů v případě nesprávně nastavených pravidel. [2]

3.4.2 Podpisové systém detekce průniku

Podpisové, nebo-li signaturové systémy detekce narušení analyzují průchodnost sítě a porovnávají pakety s databází známých atributů útoku. Tento přístup je podobný principu, na kterém funguje většina antivirového softwaru. Zde je hlavním problémem stárnutí signaturních databází – mezi vznikem nových typů útoků a aktualizací signaturné databáze může být příliš mnoho času, během něhož IDS nebude moci tuto hrozbu odhalit.

Nevýhody detekčních metod:[20]

1. Nepřijatelně vysoká míra falešných spouštění a průchodu útoků;
2. slabé možnosti detekce nových útoků;
3. většina narušení bezpečnosti, je nemožné určit, v počátečních fázích;
4. je těžké, někdy nemožné určit útočníka, cíle útoků;
5. nedostatek hodnocení přesnosti a přiměřenosti výsledků práce;
6. je nemožné definovat „staré“ útoky, které využívají nové strategie;
7. složitost detekce narušení bezpečnosti v reálném čase s požadovanou plnosti ve vysokorychlostních sítích;
8. slabá schopnost automaticky detekovat složité koordinované útoky;
9. značná přetížení systémů, ve kterých fungují IDS, při práci v reálném čase;

4 Denial-of-service attack/Distributed Denial of Service

Jednoduše řečeno, útoky DoS provádí určitou škodlivou aktivitu zaměřenou na přivedení počítačového systému do stavu, ve kterém nemůže obsluhovat uživatele nebo správně vykonávat funkce, které mu byly přiděleny. Do stavu „denial of service“ obvykle vedou chyby v SOFTWARU, nebo nadměrné zatížení na kanálovou síť nebo systém jako celek. Výsledkem je, že software nebo celý operační systém stroje „padá“ nebo se ukáže být ve „smyčkovém“ stavu. To ohrožuje prostoje, úbytek návštěvníků / zákazníků a ztráty.

4.1 Anatomie DoS útoků

DoS útoky jsou rozděleny na úrovni protokolu a úrovni aplikací(vzdálené).[18] Lokální exploity zahrnují různé druhy exploitů, Fork bomba a programy, které otevírají jeden milion souborů nebo spouští určitý cyklický algoritmus, který spotřebovává paměťové a procesní zdroje. Zde je podrobnější výpis vzdálených útoků DoS, které jsou rozděleny na dva druhy:
Vzdálený provoz chyb SOFTWARU s cílem uvést jej v nepoužitelném stavu.
Flood – zasílání velkého množství nesmyslných (méně často smysluplných) paketů na adresu oběti. Cílem floodu může být komunikační kanál nebo prostředky stroje. V prvním případě paketový tok zaujímá celou šířku pásma a neumožňuje napadenému stroji zpracovávat oprávněné požadavky. V případě druhém prostředky stroje jsou zachyceny pomocí opakovaného a velmi častého přístupu k službě, která provádí komplexní operaci náročnou na zdroje. Může to být například dlouhý přístup k jedné z aktivních komponent (skriptu) webového serveru. Server spotřebuje veškeré prostředky stroje při zpracování požadavků útočníka a uživatelé musí čekat.
V tradiční exekuci (jeden útočník – jedna oběť) zůstává účinný pouze první typ útoku. Klasický flood je zbytečný. Jednoduše proto, že se současnou šířkou serverového kanálu, úrovní zpracovatelského výkonu a rozsáhlým používáním různých anti-DoS technik v softwaru (např. zpoždění v opakovaném provádění stejných akcí jedním klientem), se útočník změní v nepříjemného komára, který nemůže způsobit žádné škody.
Distribuovaný útok Denial of Service (DDoS) používá stejný princip jako útok DoS, ale technicky je odlišný. Podstatou útoku je, jak jeho název napovídá, že na stejném místě ve stejném čase je mnoho útočníků, což v konečném důsledku vede ke stejným výsledkům, jako při útoku DoS. Provedení tohoto útoku je však obtížnější, neboť vyžaduje spoustu počítačů. A tudíž jsou k provádění tohoto útoku nejčastěji

využívány botnetové sítě.
Někdy útok DDoS nastane neúmyslně, jakmile velký počet uživatelů vstoupí na stránky pouhou náhodou. Například při oznamování malého místa na portálech s velkou návštěvností se taková stránka nemusí jednoduše vyrovnat s přílivem uživatelů a dočasně nemusí být k dispozici.
Nicméně důsledky DDoS útoků a jejich účinnost lze výrazně snížit na úkor správného nastavení router, firewall a neustálou analýzou anomálií v síťovém provozu. V další části článku budou důsledně projednány:

- způsoby rozpoznání počátku útoku DDoS;
- metody pro boj s konkrétními typy DDoS útoků;
- univerzální tipy, které pomohou připravit k DoS útoku a snížit jeho účinnost.

4.2 Flood-útoky

Existují dva typy DoS/DDoS útoků. Nejčastější z nich je založen na myšlence flood, což je zahrnování oběti velkým množstvím balíků. Flood bývají různé: ICMP-flood, SYN-flood, UDP-flood a HTTP-flood. Moderní DoS-boti mohou používat všechny tyto druhy útoků najednou, takže je třeba předem postarat se o dostatečnou ochranu každého z nich.[17] Níže jsou uvedeny příklady, jak se chránit od velmi častých typů útoků.

4.2.1 SYN-flood

Při běžné interakci s klientem a serverem odesílá SYN paket požadavek na otevření připojení. Server reaguje kombinací paketů SYN + ACK. Poté, co klient odpoví na balíček ASK, je spojení vytvořeno a klient může bezpečně použít prostředek. V případě SYN-flood, útočníci ve velkých počtech a v krátké době posílají SYN pakety, zatímco ignorují odpověď serveru. Výsledkem je, že se objeví fronta „napůl otevřených“ dotazů, které přeplní frontu připojení a neumožňují legitimním klientům posílat SYN pakety na server. Spojení oprávněného klienta se zdrojem teoreticky lze provést, ale pokud se tak stane, pak jen s velkým zpožděním.

4.2.2 UDP-flood

K dnešnímu dni tento typ flood je považován za nejméně nebezpečný, jelikož hackerské programy, které ho používají, jsou snadno zjistitelné a blokovatelné. Při použití UDP-flood, hlavním cílem není samotný zdroj a jeho komunikační kanál. V první

řadě jsou UDP-pakety přístupné poskytovateli, tudíž útočníci využívají velké množství paketů tohoto typu. Děje se tak proto, aby zabránily TCP-pakety legitimním uživatelům přístup na server.

4.2.3 ICMP-flood

Na útok využívající protocol ICMP se nejčastěji využívají pakety ICMP Echo, které pomocí ping-u zjišťují, zda je vzdálené zařízení dostupné. Tím, že útočník může zvolit velikost odesílaného ICMP paketu až do velikosti 65Kb a nastavit frekvenci odesílání, dosáhne, že datová linka cílového počítače bude značně přetížena. Navíc, pokud útočník zfalšuje adresu odesílatele, datová linka bude zaplavovaná dvojnásobně a útok tak bude ještě masivnější.

4.2.4 HTTP-flood

Tento typ je jedním z nejčastějších způsobů flood. Je založen na nekonečném odesílání HTTP zpráv GET na 80. port s cílem načíst webový server tak, aby se ukázalo, že zprávy nejsou schopny zpracovat všechny ostatní dotazy.
Tento flood útok může být zaměřen na kořenový server a jeho skript, který je zaneprázdněn při provádění úloh náročných na zdroje. Rozpoznávání detekci útoku je možné díky rychlému růstu počtu žádostí na jeden nebo více skriptů na serveru a rychlým růstem logů serveru. Některé boty mohou nejen generovat různé požadavky, ale také měnit jejich délku a datovou rychlost. To je považováno za nejlepší útok, což nepochybně přinese vysoké výsledky.[21]
Efektivní boj s flood HTTP je možný díky vylepšení serveru a databáze, vyloučení Dos-botů.[21]Konkrétní opatření zahrnují zvýšení maximálního počtu používaných souborů a připojení, použití metody epoll pro zpracování připojení, odpojení časový limit na uzavření keep-alive spojení.

4.3 DoS útoky na zranitelnosti v software na DNS serverech

Takhle se nazývají útoky na cache. Během útoku jako první nahradí útočník IP adresu serveru DNS oběti. Poté požadavek oběti na stránku HTML spadá buď do „černé díry“ (pokud byla adresa IP nahrazena neexistujícím serverem), nebo přímo na server útočníka. Druhý případ je mnohem závažnější, neboť útočník má snadný přístup k osobním údajům o nic netušící oběti. Lze uvést příklad, jak k tomu dochází. Předpokladem je, že se chce klient dostat na Web-uzel vutbr.cz. Ovšem ve

zranitelnosti v DNS serveru VUT již útočník nahradil IP adresu uzlu vutbr.cz za svoji. Nyní je oběť automaticky přesměrována na uzel útočníka.

5 Systém detekce / prevence průniku Suricata

Suricata open-source IPS/IDS systém. Velmi podobný program jako Snort. Hlavní rozdíl, jímž se liší Suricata od Snortu je možnost použití GPU v režimu IDS, pokročilejší systém IPS, multi-tasking. Suricata docela efektivně pracuje s 24 a více procesory.[11] V důsledku toho vysoký výkon umožňuje zpracovávat provoz až do 10 Gbit na běžném zařízení, a mnoho dalšího, včetně plné podpory formátu pravidel Snort. Zpočátku podporováno dekódování IPv6, včetně tunelů IPv4-in-IPv6, IPv6-in-IPv6, služba Teredo a některé další. Pro zachycení provozu se používá několik rozhraní - NFQueue, IPFRing, LibPcap, IPFW, AF_PACKET, PF_RING. Režim Unix Socket umožňuje automaticky analyzovat soubory PCAP, které byly dříve zachyceny jiným programem (např. Sniffer). Režim IDS umožňuje zkontrolovat téměř celý paket provozu, jelikož neexistuje žádné striktní omezení na dobu zpracování, a také se GPU používá k analýze. Proto je navrženo testovat provoz režimu IPS, který má teoreticky více zranitelností.[11]
V Suricata používá dva režimy IPS: NFQ a AF_PACKET.

Funkce NFQ IPS režimu:
Užitím pravidla NFQUEUE v pravidlech protokolu iptables jsou pakety odesílány do Suricata. Pokud je režim nastaven na „přijmout“, paket je odeslán do Suricata pravidlem pomocí NFQ, ve výchozím nastavení. [5]

Režim AF_PACKET na bráně:
Myšlenka je založena na jednom ze způsobů fungování snort. Ten mapuje dvě síťová rozhraní a všechny pakety přijaté z jednoho rozhraní jsou odesílány do jiného rozhraní (pokud není v balíku proveden popis s klíčem drop). Pro realizaci je nutné Suricata přidělit dvě síťová rozhraní, ovšem poskytují jen jednoduchý systém mostů. Všechna nastavení Suricata a pravidla jsou vyráběny v souborech ve formátu YAML, je grafické a zjednodušují automatické zpracování.
Jednou z výhod Suricata je zpracování 7. úrovně OSI, což zvyšuje její schopnost detekovat škodlivé programy pro aplikace. Suricata automaticky detekuje a analyzuje protokoly (IP, TCP, UDP, ICMP, HTTP, TLS, FTP, SMB, SMTP apod.),[11] takže pravidla nemusíme byt striktně vázána na číslo portu, jako tomu bylo v Snortu, stačí zadat protokol a akce.

6 Mikrotik

Mikrotikls Ltd (obchodní název MikroTik) je lotyšský výrobce počítačového síťového zařízení. MikroTik vyvíjí, vytváří a prodává drátové a bezdrátové routery, k nim operační systémy a související zařízení.

Tj. obecně se MikroTik routeros dělí na tři směry:

- RouterBoard (RB)
- RouterOS
- Školení

RouterBOARD je hardwarová platforma od společnosti MikroTik, která je v řadě směrovačích s operačním systémem RouterOS. Různé verze RouterBOARD umožňují řešit různé síťové úlohy: od jednoduchého bezdrátového přístupového bodu a spravovaného přepínače, až k výkonovému routeru s firewallem a QoS.

RouterOS podporuje mnoho služeb a protokolů, které mohou být využity středními RouterOS MikroTik nebo velkými poskytovateli jako OSPF, BGP, VPLS/MPLS. RouterOS je velmi flexibilní systém a je velmi dobře podporován Mikrotikem, jak v rámci fóra a poskytování různých Wiki materiálů, tak i specializovaných příkladů konfigurace. Po instalaci existuje velké možností následné konfigurace a dalšího řízení Mikrotik Router OS:

- MAC based – přístup k zařízení na úrovni MAC adres;
- WinBox – utilita pro Windows OS;
- WebBox – webové rozhraní;
- Webfig – rozšířené konfigurační webové rozhraní;
- Příkazový řádek (konzola) s vestavěným skriptem podporuje a pracuje na protokolech telnet a ssh;
- API – schopnost vytvářet své vlastní aplikace pro konfiguraci nebo monitorování sítě.

7 Debian

Debian je operační systém sestávající ze svobodného softwaru s otevřeným zdrojem. V současnosti je Debian GNU/Linux jednou z oblíbenějších distribucí GNU/Linux [8], která měla v primární podobě významný dopad na vývoj tohoto typu OS jako celku [7]. Existují také projekty založené na jiných jádrech: Debian GNU/Hurd, Debian GNU/kFreeBSD a Debian GNU/kNetBSD. Debian lze použít jako operační systém pro servery i pracovní stanice.

Výhody systému Debian:[19]

- Ze všech dostupných Linux je nejstabilnější.
- Je zde zvýšená pozornost věnovaná otázkám bezpečnosti.

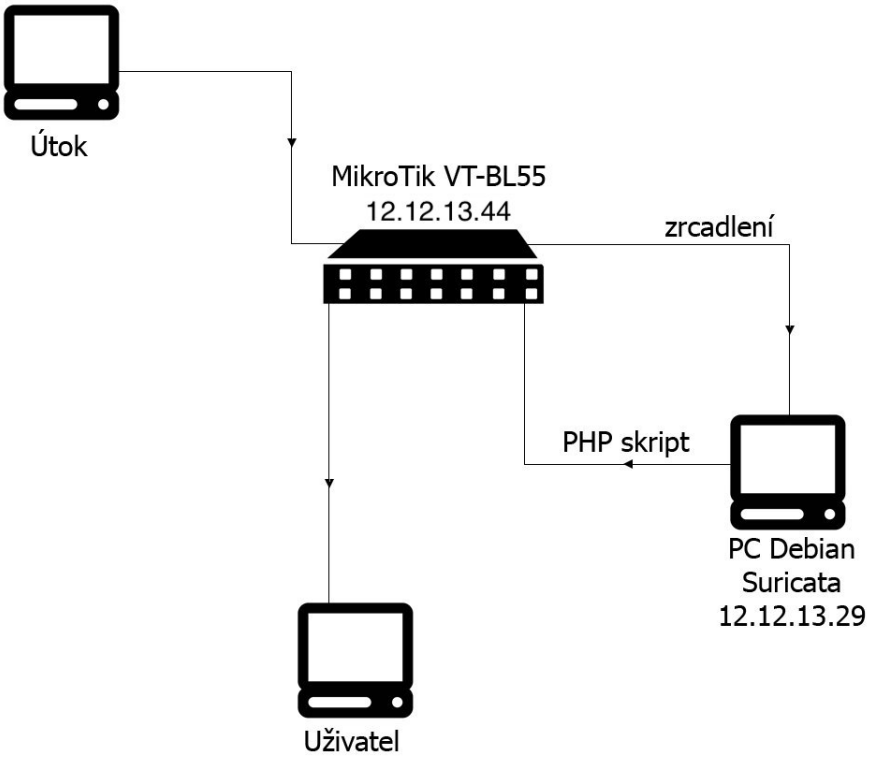
Nevýhody Debianu:[19]

- Velká mezera mezi verzemi, software se stává zastaralým.
- Mnoho věcí, které fungují v jiných distribucích z balíčku v Debianu, trvá dlouhou dobu.

8 Experimentální pracoviště

Praktická část práce byla provedena na experimentálním pracovišti s využitím infrastruktury Fakulty elektrotechniky a komunikačních technologií VUT v Brně. Experimentální pracoviště zobrazené na Obr. 8.1 je rozděleno podle přiřazení k serverům pro další filtrování operace za účelem zjištění a předání informací o navrhované operaci.

Přepínač Mikrotik před filtračními prvky rozhodne cestu k paketu podle cílové adresy IP, čímž se odesílá přes konkrétní filtrační server.



Obr. 8.1: Struktura pracoviště

9 Realizace

K realizace moje práce budu potřebovat:

1. zařízení MikroTik s operačním systémem RouterOS;
2. zařízení, na kterém poběží Suricata;
3. jeden PC na provádění testů a útoků.

9.1 Konfigurace firewallu na RouterOS

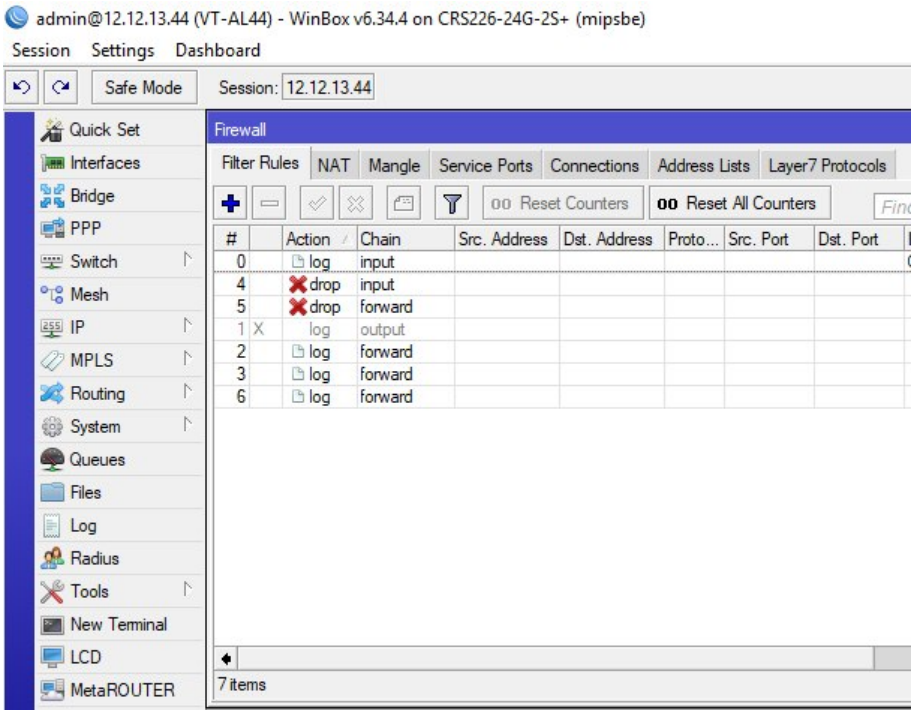
Firewall v MikroTik Routeros má velmi velké možnosti na filtrování, výběr a práci s provozem. Výběr balíčků je možný na více než 50 parametry, přičemž ke každému balíčku je možné použít jeden z několika akcí jako: accept, drop, reject, tarpit a další. Ukažme si standardní nastavení firewall Router OS[12].
Pro správné pochopení a čtení firewall v routeru potřebujeme pochopit chain průchodných paketů. V firewallu jsou tři řetězce provozu:
Input — pakety odeslané na router. Například vezměme situaci, kdy pracujeme na SSH nebo Telnet s Router OS. Celý provoz v tomto případě půjde prostřednictvím chain Input.
Forward — pakety, kteří jdou „přes“ router.
Output — pakety odeslané z routeru. Při použití přikazu Ping z routeru všechny balíčky budou generovány routerem a projdou přes chain Output.
Všechny pakety v firewall procházejí shora dolů každým pravidlem. Balík bude zpracován v firewall filter, dokud nebude spadat pod pravidlo. Místo umístění pravidel je proto velmi důležité.
Přidalá jsem se pravidla do filtrů aby přenesené IP ze skripty prošlo těmi pravidly a na základě prvního pravidla rozhodně že IP z blacklistu má zahodit:

```
/ip firewall filter add chain=input src-address-list=blacklist action=drop
/ip firewall filter add chain=forward src-address-list=blacklist action=drop
```

A když už něco zahodíme, bylo by dobrý vědět o tom něco, takže posledni pravidlo by mohlo vypadat nějak takto:

```
/ip firewall filter add chain=forward limit=5/1m,5:packet action=log
```

Pravidlo se dá slovy přeložit „jako prvních 5 paketů a pak dalších 5 během jedné minuty se zalogují“. První argument je počet paketů a za lomítkem během jaké doby. Posledním argumentem je počet paketů, které se do maximálního počtu nepočítají. Jak to vypadá můžeme vidět na Obr. 9.1



Obr. 9.1: Filter Rules

9.1.1 Přidání skripty

V MikroTiku je třeba předem vytvořit skriptu, která bude Suricata při detekci průniku příkazem vyvolávat. Tento skript přidá adresu do blacklistu a nastaví čas blockovaní na 5 min, pak adresa automatické se smaže. Vytvořila jsem prázdnou skriptu:

```
/system script add name = addIP
```

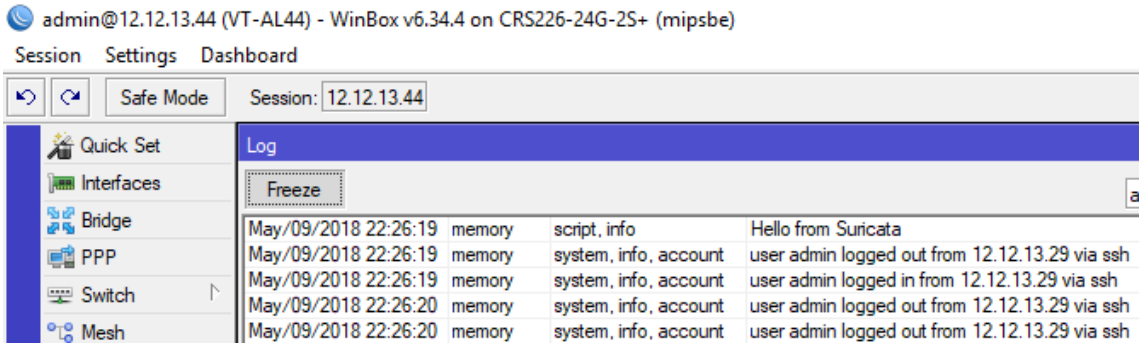
Pro zpracování skripty:

```
/system script edit [/ system script find name = addIP ] source
```

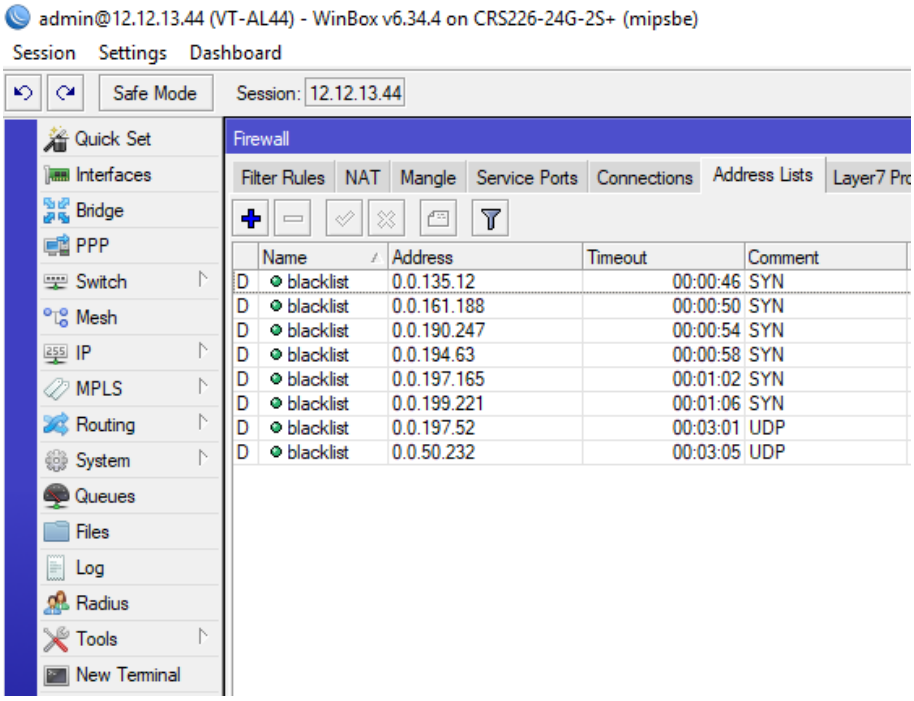
Do skripty jsem vložila:

```
log info \uv{Hello from Suricata}
:global ip;
//Přidání IP adresy do blacklistu
/ip firewall address-list add list = blacklist
address = $ip comment =$time
```

Přihlášení můžeme sledovat v Logu na stárne Mikrotiku Obr.9.2 Po přihlášení se mně začne blockovat IP adresy které přišli od Suricata a zanášet do blacklistu Obr.9.3



Obr. 9.2: Log na straně Mikrotik



Obr. 9.3: Blokovaní IP na 5 min

9.2 Vytváření skripty

Na serveru jsem vytvořila PHP skript, nazvala jej „LogChecker.php“, a poté přesunula do složky „//home/toor“. Každou minutu Skript projde obsah fast.logu a hledá, jestli za tu dobu Suricata nezaznamenal útok. Pokud najde útok, tak pošle poslední čas do skripty kterou taky vytvořila pro porovnání posledních příchozích události „LastDateInfo.dt“ a srovnání s fast.log a když data přidají připojí se na MikroTik.9.2

9.2.1 Automatizace skripty

Posledním krokem bylo spuštění PHP skripty každou minutu. To poskytne tzv. „Crontab“, který se bude pravidelně spouštět. Crontab pracuje jako plánovač, kde lze nastavit počet spuštění příkazů za zvolenou jednotku času. V systému Debian jsem otevřela editační prostředí pro crontab:

```
crontab -e
```

Na konec souboru jsem vložila:

```
*/1 * * * * /home/toor/LogChecker.php
```

Zatím provedla restartování:

```
service cron reload
```

Pak pro spouštění potřebovala ještě povolit pro skript práva root příkazem:

```
chmod 700 LogChecker.php
```

Crontab mi začal v minutovém intervalu spouštět můj PHP script.

9.2.2 Generování DoS útoků

Mám k dispozicí dva tipy DoS útoky SYN a UDP které se naházejí přímo na serveru ve složce „/home/toor/pcap/“. Spuštění musí být provedeno přímo z konzoly. Následující příkaz odpovídá podmínkám, a obsahuje vše aby zahájit detekcí:

```
suricata -c /etc/suricata/suricata.yaml -r /home/toor/pcap/syn.pcap
suricata -c /etc/suricata/suricata.yaml -r /home/toor/pcap/udp.pcap
```

Otevřela ještě jeden terminál pro sledování kam se útoky zapisují. Pomocí příkazu „tail“ otevřeme poslední log, charakteristickým znakem příkazů „tail“ je jeho schopnost nekončit, když je dosaženo konce výstupu, ale očekávat vznik nových dat to provedeme, přidělením přepínače -f:

```
tail -f /var/log/suricata/fast.log
```

Struktura fast.log
Jednotlivé výstrahy manuálně nastavené v local.rules jsou na obrázcích níže. Každá jedna výstraha obsahuje datum, nastavené, zprávu, prioritu zdrojovou a cílovou IP adresu.

```
11/11/2017-16:22:05.632816  [**] [1:5:0] Unusually fast port 80 SYN packets inbound [**]
[Classification: Misc activity] [Priority: 3] {TCP} 0.0.39.217:15914 -> 192.168.2.160:80
```

Obr. 9.4: Výstraha Suricata po SYN flood útoku

```
11/11/2017-16:25:45.571621  [**] [1:555:0] Unusually fast UDP packets inbound [**]
[Classification: Misc activity] [Priority: 3] {UDP} 0.0.35.99:42077 -> 192.168.2.160:80
```

Obr. 9.5: Výstraha Suricata po UDP flood útoku

```
11/14/2017-15:08:08.550141  [**] [1:2100486:5] GPL ICMP_INFO Destination Unreachable
Communication with Destination Host is Administratively Prohibited [**] [Classification:
Misc activity] [Priority: 3] {ICMP} 195.181.212.55:3 -> 12.12.13.29:10
```

Obr. 9.6: Výstraha Suricata po Ping of Death útoku

Meta klíčová slova
Meta-nastavení nemají žádný vliv na kontrolu společnosti Suricata, mají vliv na to, jak Suricata hlásí události.
Na obrázku vidím klasické klíčové slovo [**Classification: Misc activity**] které poskytuje informace o upozornění. Skládá se z krátkého jména, a dlouhého jména. Může říkat například, zda je pravidlo pouze informativní, nebo že je o hackerů a tak dále [11].

Tak že je tam máme prioritní klíčové slovo [**Priority: 3**] které obsahuje povinnou číselnou hodnotu, která se může pohybovat od 1 do 255. Nejčastěji se používají čísla od 1 až do 4. Nejdříve budou zkoumány podpisy s vyšší prioritou. Nejvyšší prioritou je 1 [11].

9.2.3 Připojení na Mikrotik SSH

Potřebovala jsem vazbu na knihovnu »libssh2, která poskytuje přístup ke zdrojům (shell, vzdálený start, tunelování, přenos souborů) na vzdáleném počítači za použití

kryptograficky chráněného přenosu dat [13].
Zistila jsem že uvedená knihovna je v systému, příkazem:

```
apt-cache search openssl-dev
```

Pro spijení jsem použila funkce SSH2:

`ssh2_connect` - Naváže spojení se serverem přes SSH.
`ssh2_auth_password` - Autentizace přes SSH pomocí klasického hesla.
`ssh2_exec` - Spuštění příkazu na vzdáleném serveru a výběr pro ni kanálu.

10 Testování práce

10.1 Graphing — nástroj pro monitorování v RouterOS

V operačním systému RouterOS je grafický nástroj Graphing umožňuje sledovat různé parametry routeru MikroTik a jejich zobrazení v grafech.[14]
Nástroj může zobrazovat následující informace v grafech:

- Napětí a teplotu Routerboard;
- Využití CPU, paměti, využití disku;
- Provoz na síťových rozhraních;
- Provoz ve frontách, simple queues.

Grafing se skládá ze dvou částí: první část shromažduje informace, zatímco druhá zobrazuje grafiku na webové stránce.
Chcete-li zobrazit grafy, zadejte v prohlížeči adresu typu „http:// IP adresa routeru/graphs/„

Pro zobrazení grafů:

```
/tool graphing interface add interface=GE17
```

Ve výchozím nastavení Graphing nezobrazuje grafiku pro využití procesoru, paměti nebo disku. K dispozici jsou pouze odkazy na grafiku síťového rozhraní.
Pro zobrazení grafy využití cpu, paměti a disku:

```
/tool graphing resource add allow-address=0.0.0.0/0
```

Data ze směrovače se shromažďují každých 5 minut, ale ukládají se na jednotku systému každý Every Store. Po restartování směrovače se grafy zobrazují před posledním restartováním informací, které byly na disku uloženy.[14]

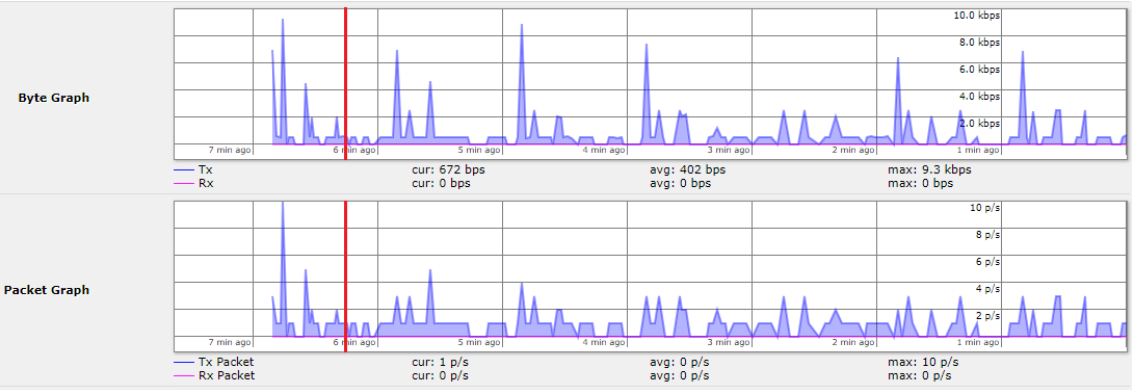
```
/tool graphing set store-every=5min
```

11 Výsledky testování

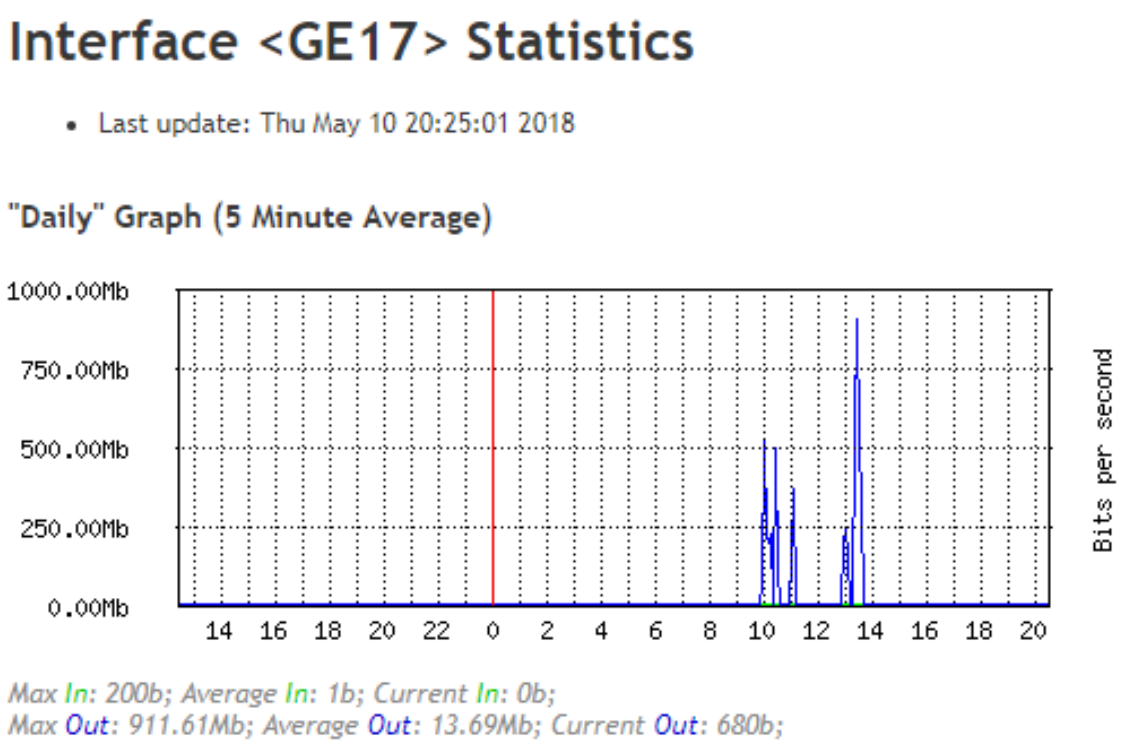
11.1 Distributed Denial of Service SYN

Testování jsem provedla postupním zvyšováním DDoS útoků, SYN útok jsem testovala 5 hodin je možno vidět, že k pronikání došlo, přenosová rychlost byla 150 - 900 Mb/s. Vyšší hodnoty tento útok nedosáhl.

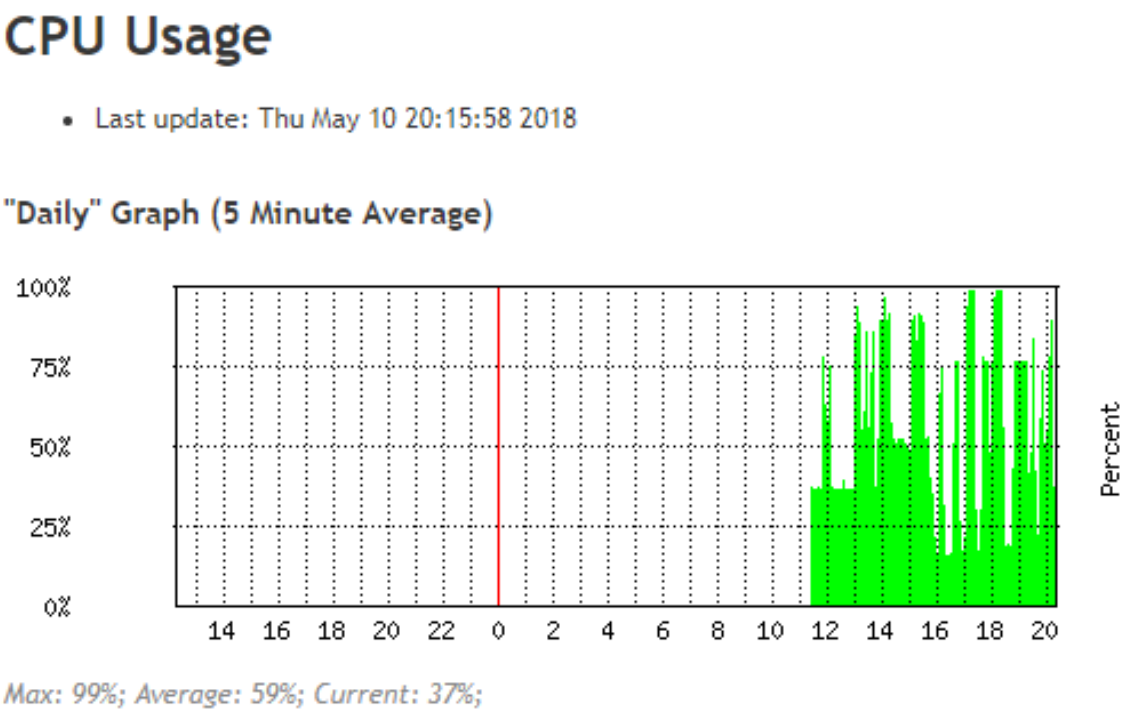
Na obrázku 11.1 znázorněno začátek SYN útoku na rozhraní GE17 za prvních 5 min, na hoře se znázorněno Bajt Graf střední odchozí provoz je činí 402 bps, dole je paket graf u kterého odchozí provoz je 0p/s.
Na obrázku 11.2 zobrazeno testování za 3 hodiny, ale skutečně jsem prováděla testování SYN 5 hodin, ale potom už byl nějaký výpadek a přestalo se mně zobrazovat další 2 hodiny, tak že bylo třeba restartovat grafing na Interfacu GE17, pro další testování UDP. Dokonce můžeme vidět výsledek že střední odchozí provoz je 13,69Mb, a kvůli tomu výpadku, vidíme malou střední hodnotu, při tom že mněla by být najakych cca 60Mb.
Na obrázku 11.3 vidím CPU využití na straně Mikrotik, tady a v dalších obrázcích mne už normálně zobrazovalo těch 5 hodin, a střední využití bylo 59% při spuštění detekce pro syn flood útok cca desetkrát.
Na obrázku 11.4 zobrazeno paměťové využití které má celkem k dispozici 64Mib při SYN flood útoků bylo maximální využití paměti je 30,92MiB a střední je 29,50MiB což lze o tom říct že bylo použíta půlka paměti.
Na obrázku 11.5 zobrazeno využití disků které má celkové místo na disku 128MiB z toho při útoku bylo použíto cca 14%



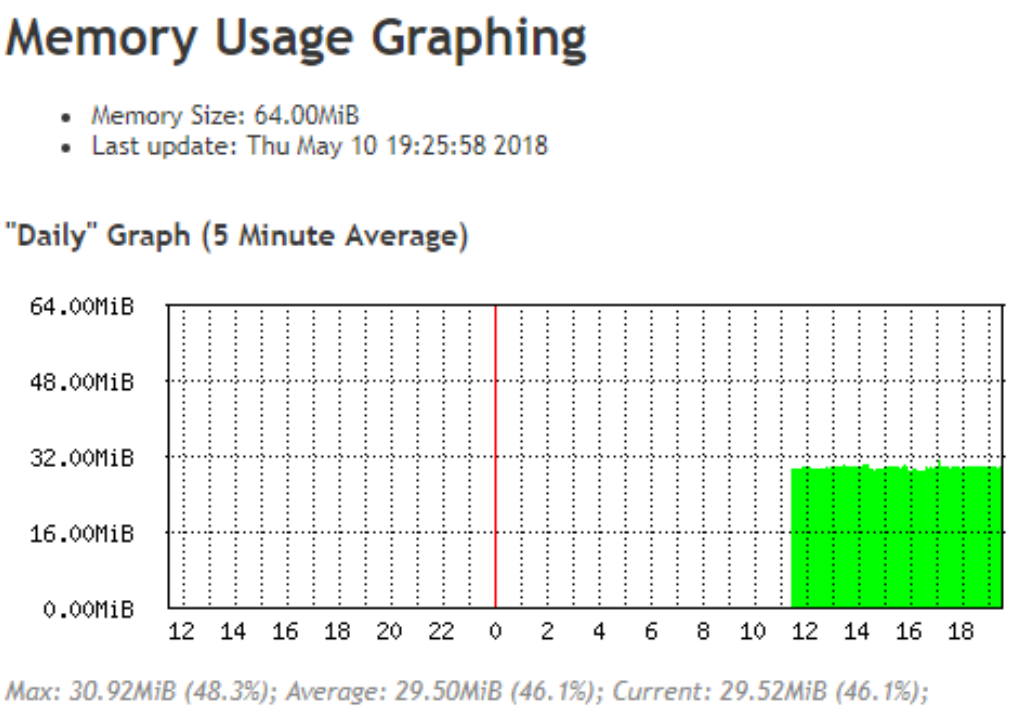
Obr. 11.1: Rozhraní provozu GE17 za 5min



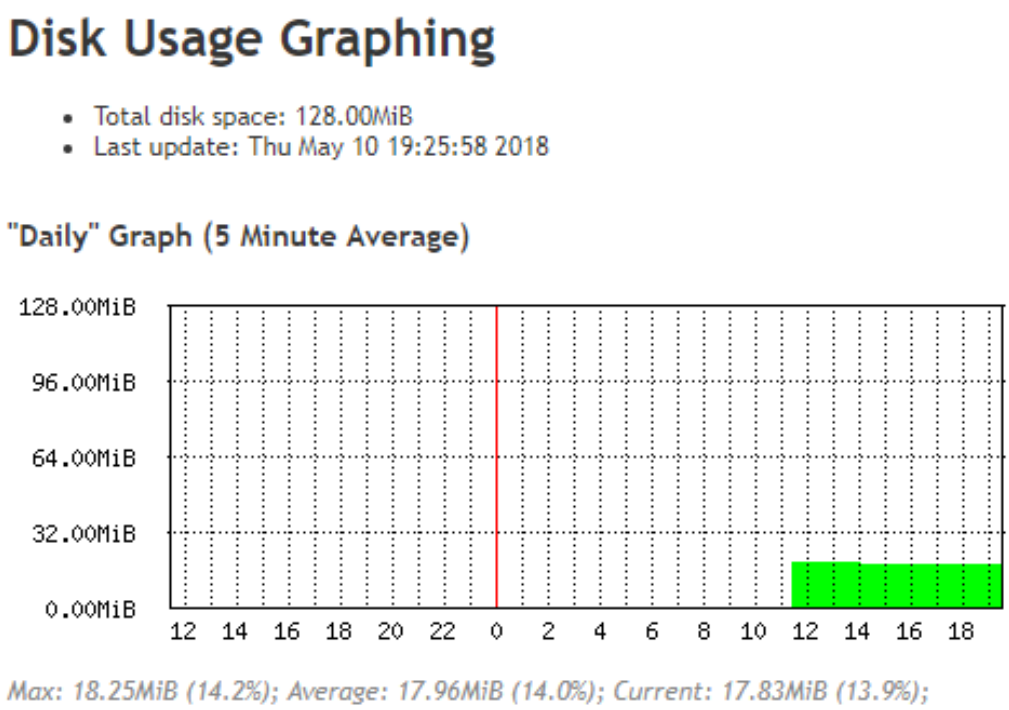
Obr. 11.2: Rozhraní GE17 statistiky za 5 hodin při zvyšovaní DDoS utoků



Obr. 11.3: CPU využití za 5 hodin při zvyšovaní DDoS utoků



Obr. 11.4: Graf paměťové využití za 5 hodin při zvzšovaní DDoS utoků

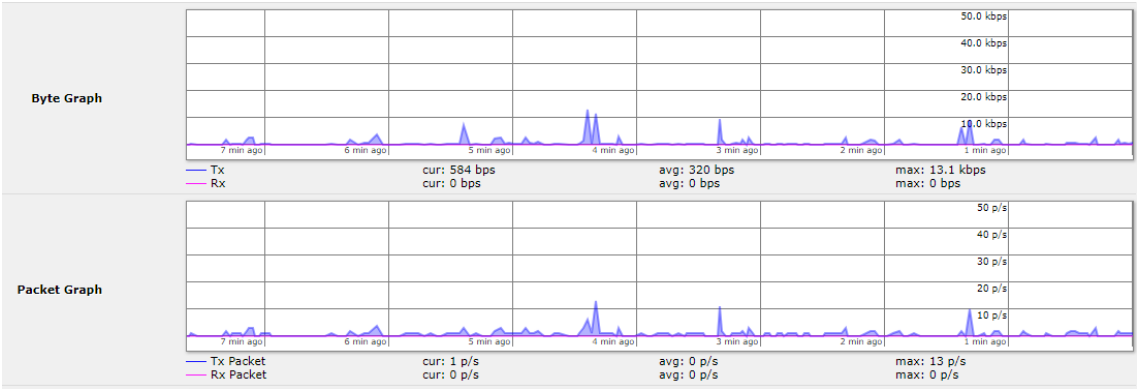


Obr. 11.5: Graf využití disku za 5 hodin při zvyšovaní DDoS utoků

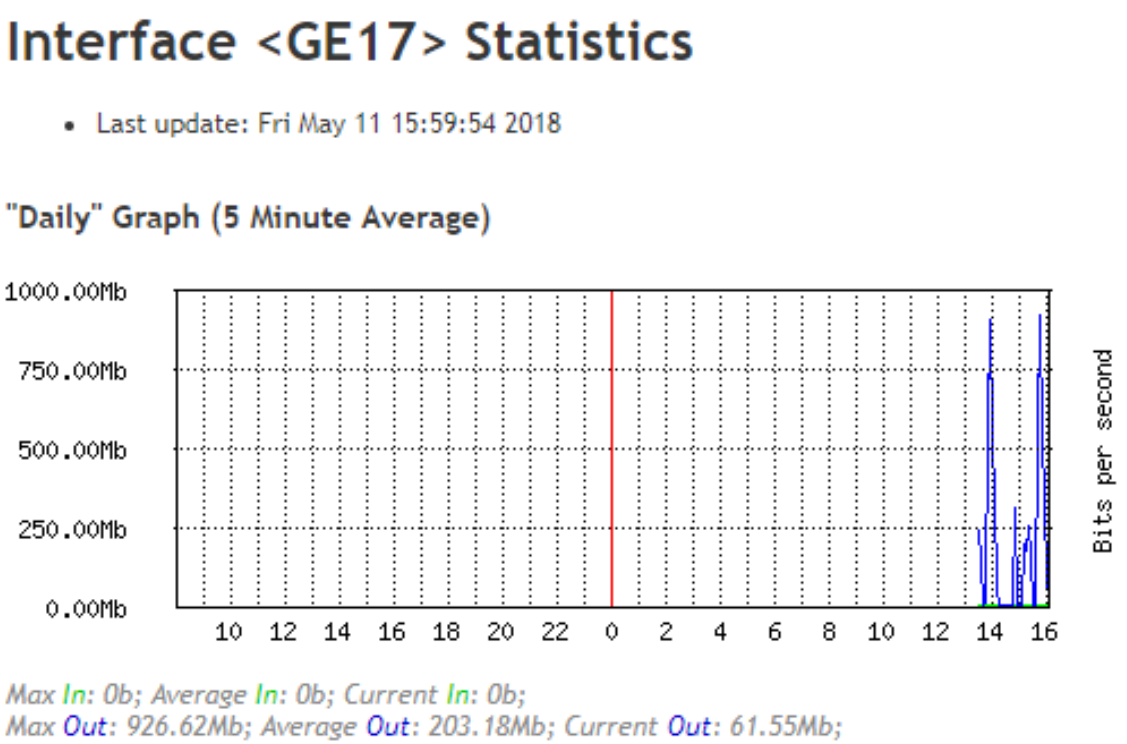
11.2 Distributed Denial of Service UDP

Testování UDP jsem prováděla 2 hodiny postupným zvyšováním na cca šestkrát s důvodů rychlého vyčerpání komunikačního kanálu.

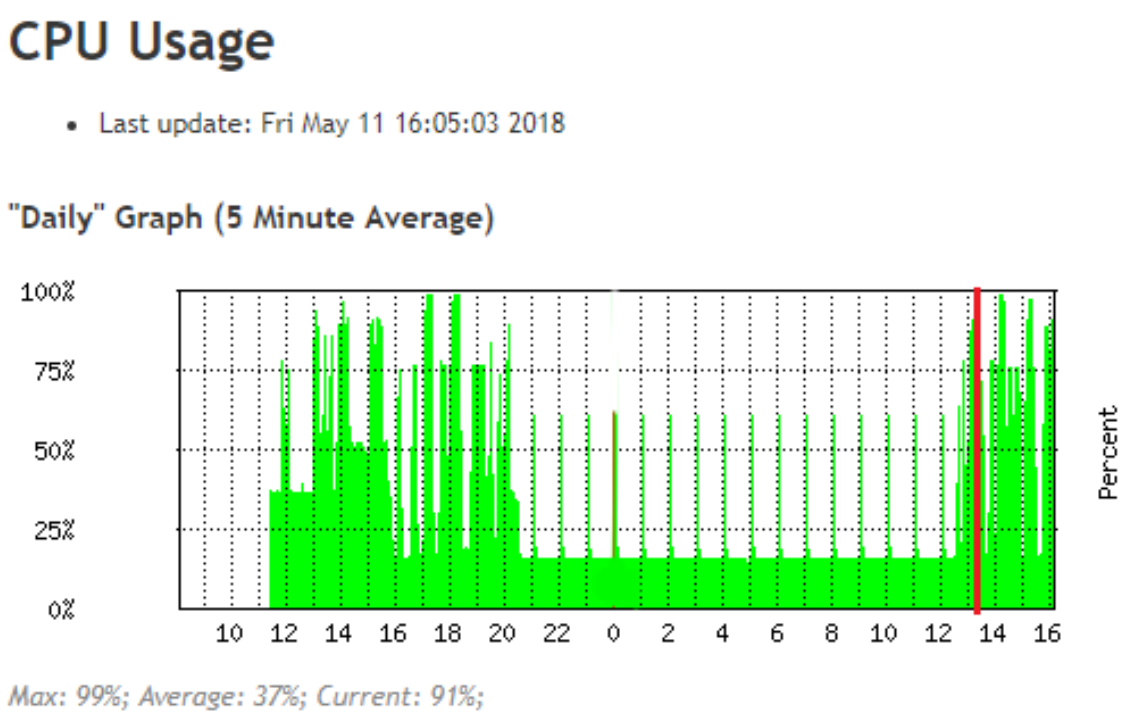
Na obrázku 11.6 zobrazen normali stav interfacu GE17 za prvních 5 min, při UDP útoku střední odchozí provoz nabýval se do 320bps, ten to graf mně posloužil pro porovnání provozu s grafem na obrázku 11.7
Na obrázku 11.7 byl zachycen odchozí provoz z Interfacu GE17 u kterého je statisticky střední hodnota je 203,18Mb odsud můžeme zhodnotit jaký je velký provoz v porovnání s obrázkem 11.6, je to kvůli tomu že UDP útoky mají velkou velikost balíku.
Na obrázku 11.8 zobrazen Graf i s SYN útoky protože nešlo mně vynulovat, a i přes to jde sledovat podle momentálního času že je využito až 91% ze 100% při UDP útoku.
Na obrázku 11.9 zobrazen graf paměťového využití který zůstává stejný jako i při SYN útoky, což činí půlku paměti.
Na obrázku 11.10 zobrazen graf využití disku který zůstává stejný jako i při SYN útoky, a to je maximální 14.2%.



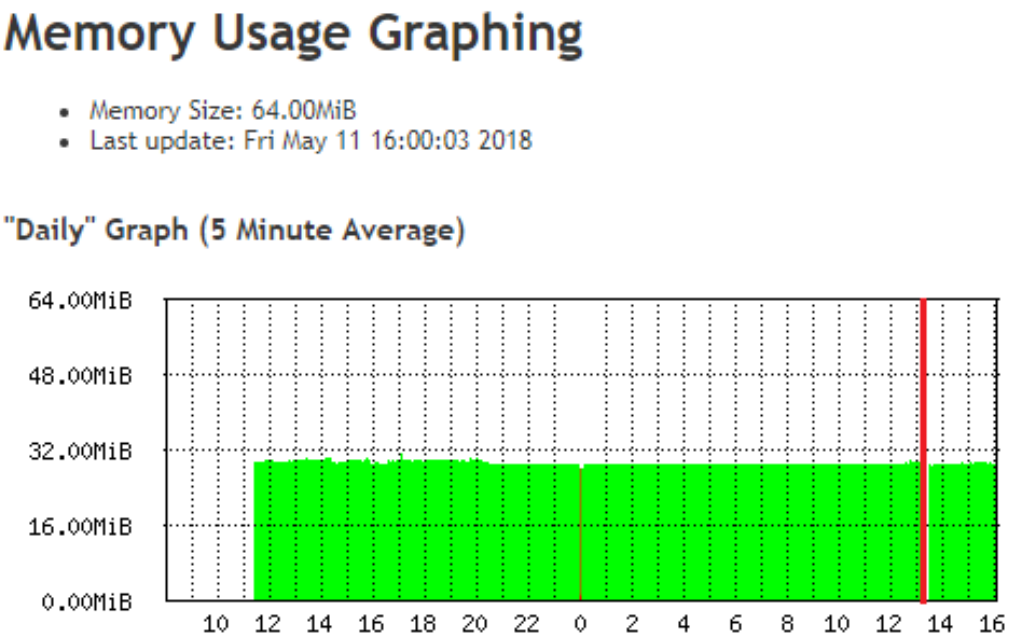
Obr. 11.6: Rozhraní provozu GE17 za 5min



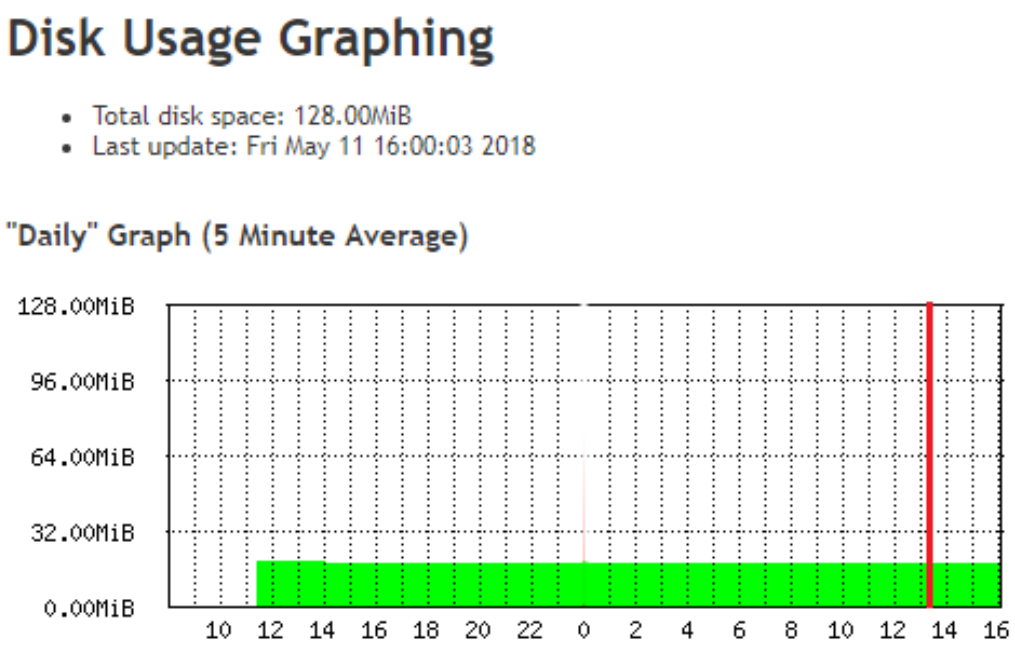
Obr. 11.7: Rozhraní GE17 statistiky za 2 hodiny při zvyšovaní DDoS utoků



Obr. 11.8: CPU využití za 2 hodiny při zvyšovaní DDoS utoků



Obr. 11.9: Graf paměťové využití za 2 hodiny při zvzšovaní DDoS utoků



Obr. 11.10: Graf využití disku za 2 hodiny při zvyšovaní DDoS utoků

11.3 Porovnání SYN a UDP

Z grafy na obrazcích 11.2 a 11.7 můžeme pozorovat že SYN útok zatížit interface jenom průměrně o 13,69Mb/S ve srovnání z UDP ktery má průměrně 209,18Mb/s. tim je možno vidět, že k pronikání došlo, přenosová rychlost byla 150 - 900Mb/s. Zatížení CPU u SYN je 37% a u UDP je 91% je to tím že UDP má mnohem větší velikost packetu než SYN. Tyto útoky byly objevovány a poslány od jedné IP adresy. Suricata tyto útoky úspěšně detekovala a upozornila hlášením. Automaticky byl spouštěn na servru PHP skript po tomto zpracované hlášení odeslal s IP adresou útočníka. Adresa útočníka byla přidána do blacklistu a komunikace byla zablokována. Blokování proběhalo rychle cca 20-30 sekund. Tehdy když během velké množství přihlášení do Mikrotik dochází k zatížení sítě a tím příchozí adresa už nezapisují se do blacklistu.

12 Závěr

Seznámila jsem se s různými typy internetových útoků a jejich vliv na počítačové sítě a koncových uživatelů. Dozvěděla jsem o detekce různých typů DDoS útoku, kam nájdené Suricatou útoky zapisují, jakou prioritu přidává různým typům útoků. V rámci bakalářské práce jsem vytvořila na serveru skriptu PHP. První co jsem udělala tak je to bylo spojení s Mikrotikem, pak jsem napsala skript pro blokování IP adresy, ale mně to na začátku stálo blokovat všechny IP adresy které je v souboru, tak že pak jsem potřebovala aby mně se blokovali jenom aktuální vyhlášky, na toto jsem potřebovala prostudovat jak ty útoky se zapisují jakou mají strukturu, podle čeho se dá stanovit že je to DDoS-flood útok a zjistila jsem že mnění se pouze podle času, potřebovala napsat kousek kódu aby byla schopna identifikovat vyhlášky podle sekund, ale potom jsem musela vytvořit soubor kdy ty poslední zachycené sekundy by se ukládali, pro další porovnání s hlavním souborem kdy ty útoky se objevují. Skript mně začal pracovat a posílát IP adresy a přidávát jejch do blacklistu na 5 min. Aby se mně tyto útoky zapisovali se do blacklistu a nastavil se tam čas, potřebovala vytvořit příkaz který by se mně to umožnil na straně Mikrotiku. Když už věděla kterým příkazem můžu přidávat IP adresy, tak zbylo mně jenom najít umístění odkud se bude vyvolávat, tak že to byl Skript na straně Mikrotiku, do kterého volá PHP skript ze serveru. Ještě jsem mněla za úkol poslat k tomu komentář o který typ útoku se jedná od skripty PHP na serveru, ale toto se mně nepodařilo, tak že jsem to vyřešila tak že vytvořila skript který je na straně Mikrotiku a přidává komentář. Svůj skript jsem otestovala dvěma typy útoků SYN a UDP. A provedla jsem porovnání na základě výsledcích mezi těmi to flood útoky. Změřila přetížení na rozhraní, jaké mněli vlivy na CPU, paměťové využití a využití disku.

Literatura

[1] NEWMAN, Robert C. *Computer security: protecting digital resources.* Sudbury, Mass.:Jones and Bartlett Publishers, c2010. ISBN 9780763759940.

[2] DEBAR, Hervé, Ludovic ME a S. Felix WU. *Recent advances in intrusion detection: third international workshop, RAID 2000, Toulouse, France, October 2-4, 2000:proceedings.* New York: Springer, 2000. ISBN 3-540-41085-6.

[3] VACCA, John R. *Computer and information security handbook.* Third edition. Cambridge, MA:Morgan Kaufmann Publishers, an imprint of Elsevier, 2017. ISBN 9780128038437.

[4] ANDERSON, Ross. *Security engineering: a guide to building dependable distributed systems.* 2nd ed. Indianapolis, IN:Wiley Pub., c2008. ISBN 978-0-470-06852-6.

[5] BOSWORTH, Seymour., Michel E. KABAY a Eric WHYNE. *Computer security handbook.* Sixth edition. Hoboken, New Jersey:John Wiley and Sons, 2014. ISBN 978-1-118-13410-8.

[6] PETROVIČ, Michal a Michal KOSTĚNEC. *Bezpečnost počítačových sítí.* Plzeň: Západočeská univerzita v Plzni, 2012. ISBN 978-80-261-0117-8.

[7] Software distributions based on Debian <`https://en.wikipedia.org/wiki/List_of_Linux_distributions`>.

[8] Debian <`https://www.debian.org/derivatives/`>.

[9] WHITMAN, Michael E. a Herbert J. MATTORD. *Principles of information security.* 4th ed. Boston, MA:Course Technology, c2012. ISBN 1111138214.

[10] ZHENG, Jun. *Ad Hoc networks: first international conference, ADHOCNETS 2009, Niagara Falls, Ontario, Canada, September 22-25, 2009:revised selected papers.* New York:Springer, c2010. ISBN 978-3-642-11722-0.
Dostupné v URL
<`http://www-users.cs.york.ac.uk/~jac/PublishedPapers/AdhocNetsFinal.pdf`>

[11] Suricata User Guide <`http://suricata.readthedocs.io/en/latest/`>.

[12] RouterOS <`https://spw.ru/educate/articles/routeros/`>.

[13] Manual PHP <`http://php.net/manual/en/intro.ssh2.php`>.

[14] Mikrotik <https://mikrotik.com/testdocs/ros/2.9/tools/graphing.php>.

[15] Přednáška <https://www.intuit.ru/studies/courses/4088/1286/lecture/24234>.

[16] Detekce narušení <https://it-sektor.ru/obnaruzhenie-vtorzheniyi.html>.

[17] DDoS <https://habr.com/company/ua-hosting/blog/233903/>.

[18] DoS <http://farmaz0n.blogspot.cz/2009/11/?m=0

[19] Debian <https://www.debian.org/intro/why_debian.en.html>.

[20] Nevýhody detekčních metod <http://citforum.ru/security/internet/ids_overview/>.

[21] HTTP-flood <https://cosmonova.net/page/DDos-attack>.

Seznam symbolů, veličin a zkratek

DNS	Domain Name System
DDoS	Distributed Denial of Services
DoS	Denial of Service
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protoco
ICMP	Internet Control Message Protocol
IDP	Insurance Data Processing
IDS	Intrusion Detection System
IP	Internet Protocol
IPS	Intrusion Prevention System
IPv6	Internet Protocol version 6
MAC	Media Access Control
NIDS	Network Instrusion Detection System
NIPS	Network Instrusion Prevention System
OISF	Open Information Security Foundation
OSI	Open Systems Interconnection
QoS	Quality of Service
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
VPN	Virtual Private Network

Seznam příloh

A Příloha

```
<?php

$blocked=array();
$fastlog = "cat␣/var/log/suricata/fast.log"; //přijde do fast.log
exec($fastlog, $lastMin); //spustí externí program
foreach($lastMin as $line)
{
$line = strtolower($line); //nebude všímat velké nebo malé písmena
preg_match_all("/([0-9]{2}:){2}[0-9]{2}/", $line, $NewDateString);
//najde čas který se změnil
$aLastDateString = split(":",file_get_contents("/var/log/suricata
/LastDateInfo.dt"));
$aNewDateString = split(":", $NewDateString[0][0]);

if($aNewDateString[1] > $aLastDateString[1] ||
(($aNewDateString[1] == $aLastDateString[1]) && ($aNewDateString[2]
> $aLastDateString[2])))
{
for($priority = 2; $priority <= 5; $priority++)
{
if (preg_match("/priority:␣?".$priority."/", $line) === 1
||
preg_match("/misc␣activity/", $line) === 1)
//jestli ve stroke bude misc activity nebo priorita vyšší
než jednička
{
preg_match_all("/\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}/", $line,
$matches);
//najde první IP adresu ve řádku
$filter=$matches[0]; //poměšti její do masivu
if (!in_array($filter, $blocked))
{
file_put_contents("/home/toor/LastDateInfo.dt", $NewDateString[0]);
$blocked[]=$filter;
sendMikrotik('12.12.13.44', 'admin', 'ijmilyjezko',$filter[0]);
//pošle potřebné údaje
}
```

```
}
}
}
}

function sendMikrotik($mt,$user,$pass,$filter)
//pomocí této funkce připojí se na Mikrotik
{
$connection = ssh2_connect($mt);
ssh2_auth_password($connection,$user,$pass);
$stream = ssh2_exec($connection, ':global␣ip␣'.$filter);
$stream = ssh2_exec($connection, '/system␣script␣run␣addIP');
//spouští se script na straně Mikrotiku
$stream = ssh2_exec($connection, 'quit');
?>
```

B Obsah přiloženého CD

Na přiloženém médiu jsou:
xgolov00.pdf.....Elektronická verze práci
LogChecker.php.....PHP Skript